



INSTITUTO  
FEDERAL  
Rio Grande  
do Sul

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

# Desenvolvimento de Aplicativo WEB I

Prof. Me. Cleber Schroeder Fonseca

Curso de Tecnologia em Análise e Desenvolvimento de Sistemas  
IFRS Rio Grande

2024



## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade



## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade

- As variáveis em PHP podem se fazer presentes em três escopos, sendo que um deles é de criação exclusiva do interpretador PHP.
- São elas:
  - superglobais
  - globais
  - locais

# Escopo de variáveis - Superglobais

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- As variáveis superglobais são criadas pelo próprio interpretador PHP, isso que dizer que não podemos alterar o valor delas durante a execução do programa.
- Cada uma delas armazena algumas informações a respeito da requisição que foi feita, sobre o servidor que está rodando a aplicação, sobre os cookies.
- Alguns exemplos de variáveis superglobais:
  - `$_GET`
  - `$_POST`
  - `$_REQUEST`
  - `$_SESSION`
  - `$_COOKIE`
  - `$_FILES`
  - `$_SERVER`
  - `$GLOBALS`

# Escopo de variáveis - Globais

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- Esse é o tipo de variável que estamos criando até este momento.
- Elas ficam disponíveis até o fim da execução da aplicação.
- Depois da criação da variável pode-se usa-la a qualquer momento dentro da aplicação.

# Escopo de variáveis - locais

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- São variáveis que só existem dentro de um trecho do código (funções).
- Assim que se encerra a execução da função a variável é excluída da memória do servidor.



① Escopo de variáveis

② Funções

③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

④ Validação de formulário

⑤ Atividade



- Existem diversos paradigmas de programação:
  - Programação funcional
  - Programação procedural
  - Programação orientada a objetos
  - Programação paralela
- Até aqui na nossa disciplina nós trabalhamos com código dito que é um tipo de programação procedural.
- Agora vamos ver como o PHP permite mudar um pouco esse paradigma e trabalhar com programação estruturada.

- O PHP permite a criação de funções para os mais diversos fins.
- A sintaxe para a criação de uma função é:

```
1 <?php
2 function nome_da_funcao (parametros) {
3     ... linhas de codigo da funcao ...;
4     [return <expressao>];
5 }
6 ?>
```

- Vamos ver um exemplo da criação de uma função.
- Nesse caso vamos criar um função que recebe dois números como parâmetro e retorna a soma desses números.

```
1 <?php
2 function somador ($val1, $val2) {
3     $soma = $val1 + $val2;
4     return $soma;
5 }
6 ?>
```

- Porém somente criar a função não faz nada, visto que no momento da criação essa função é armazenada na memória do servidor aguardando a chamada.
- Para utilizar a função deve-se utilizar a seguinte sintaxe.

```
1  <?php
2  function somador ($val1, $val2) {
3      $soma = $val1 + $val2;
4      return $soma;
5  }
6
7  $val = somador(3, 5);
8
9  echo "O resultado da soma de 4 e 8 é ",somador(4, 8);
10 ?>
```

- Para utilizar variáveis globais dentro da função pode-se utilizar a palavra-chave *global*.

```
1  <?php
2  $a = 1;
3  $b = 2;
4
5  function Soma()
6  {
7      global $a, $b;
8
9      $b = $a + $b;
10 }
11
12 Soma();
13 echo $b;
14 ?>
```

- Existem alguns casos onde podemos chamar a função dentro da própria função (recursividade).

```
1 <?php
2 function recursion($a)
3 {
4     if ($a < 20) {
5         echo "$a\n";
6         recursion($a + 1);
7     }
8 }
9 ?>
```



① Escopo de variáveis

② Funções

③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

④ Validação de formulário

⑤ Atividade

- O PHP define um conjunto de funções pré-determinadas, que podem ser encontradas em `http://www.php.net`





## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade

# Funções para debug de código

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- As funções de debug são funções que auxiliam na visualização de problemas ocorridos com variáveis
- Algumas dessas funções são:
  - void **var\_dump(\$var)**
  - void **print\_r(\$array)**

- A função **var\_dump(\$var)**<sup>1</sup> imprime na tela o tipo de dado e o valor contido na variável passada como parâmetro

```
1 <?php
2 var_dump(3.1);           // float(3.1)
3 var_dump(FALSE);        // bool(false)
4 var_dump("Texto de teste"); // string(14) "Texto de teste"
5 ?>
```

---

<sup>1</sup>[https://www.php.net/manual/pt\\_BR/function.var-dump.php](https://www.php.net/manual/pt_BR/function.var-dump.php)

- A função **print\_r(\$array)**<sup>2</sup> imprime na tela todo o array de forma ordenada, se utilizar a tag HTML

**<pre> ... </pre>**

os dados mostrados pela função ficam muito mais fáceis de serem lidos

```
1 <?php
2 $a = array ('a' => 'apple', 'b' => 'banana', 'c' => array ('x', 'y', 'z')); (
3 echo "<pre>";
4 print_r ($a);
5 echo "</pre>";
6 ?>
```

Array

```

[a] => apple
[b] => banana
[c] => Array
(
    [0] => x
    [1] => y
    [2] => z
)
)

```

<sup>2</sup>[https://www.php.net/manual/pt\\_BR/function.print-r.php](https://www.php.net/manual/pt_BR/function.print-r.php)

## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade

# Funções para arrays

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- Essas são algumas das funções para manipulação de Arrays
  - Array **array\_reverse(\$array)**
  - Array **array\_slice(\$array)**
  - Array **sort(\$array)**
  - Bool **in\_array(\$procurado, \$array)**
  - Int **count(\$array)**

# array\_reverse()

- A função **array\_reverse(\$array [, \$preserve\_key])**<sup>3</sup> retorna um novo array com os dados contidos nele invertidos
- Além do array pode ser passado um parâmetro booleano (true ou false) que tem por finalidade preservar as chaves do array, por padrão assume o valor false

```

1 <?php
2 $input = array("php", 4.0, array("green", "red"));
3 $reversed = array_reverse($input);
4 $preserved = array_reverse($input, true);
5
6 print_r($input);
7 print_r($reversed);
8 print_r($preserved);
9 ?>

```

```

Array
(
    [0] => php
    [1] => 4
    [2] => Array
        (
            [0] => green
            [1] => red
        )
)

Array
(
    [0] => Array
        (
            [0] => green
            [1] => red
        )
    [1] => 4
    [2] => php
)

Array
(
    [2] => Array
        (
            [0] => green
            [1] => red
        )
    [1] => 4
    [0] => php
)

```

<sup>3</sup>[https://www.php.net/manual/pt\\_BR/function.array-reverse.php](https://www.php.net/manual/pt_BR/function.array-reverse.php)

- A função **array\_slice(\$array, \$offset[, \$length[, \$preserve\_key]])**<sup>4</sup> retorna a sequência de um array delimitado pelos parâmetros offset e length

```

1  <?php
2  $input = array("a", "b", "c", "d", "e");
3
4  $output = array_slice($input, 2);      // returns "c", "d", and "e"
5  $output = array_slice($input, -2, 1);  // returns "d"
6  $output = array_slice($input, 0, 3);   // returns "a", "b", and "c"
7
8  // note the differences in the array keys
9  print_r(array_slice($input, 2, -1));
10 print_r(array_slice($input, 2, -1, true));
11 ?>

```

```

Array
(
    [0] => c
    [1] => d
)
Array
(
    [2] => c
    [3] => d
)

```

<sup>4</sup>[https://www.php.net/manual/pt\\_BR/function.array-slice.php](https://www.php.net/manual/pt_BR/function.array-slice.php)



- A função **sort(\$array[, \$sort\_flag])**<sup>5</sup> ordena um array, os elementos são ordenados do menor para o maior ao fim da execução
- \$sort\_flag aceita alguns valores específicos como pode ser visto no manual do php os mais comuns são:
  - **SORT\_REGULAR** - compara os itens normalmente (não muda o tipo), é o valor padrão quando nada é informado
  - **SORT\_NUMERIC** - compara os itens numericamente
  - **SORT\_STRING** - compara os itens como strings

---

<sup>5</sup>[https://www.php.net/manual/pt\\_BR/function.sort.php](https://www.php.net/manual/pt_BR/function.sort.php)

# sort()

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

```
1 <?php
2
3 $fruits = array("lemon", "orange", "banana", "apple");
4 sort($fruits);
5 foreach ($fruits as $key => $val) {
6     echo "fruits[" . $key . "] = " . $val . "<br>";
7 }
8
9 ?>
```

```
fruits[0] = apple
fruits[1] = banana
fruits[2] = lemon
fruits[3] = orange
```

- A função **in\_array(\$procurado, \$array [, \$strict])**<sup>6</sup> essa função retorna true caso o valor de \$procurado esteja dentro do array

```
1 <?php
2 $os = array("Mac", "NT", "Irix", "Linux");
3 if (in_array("Irix", $os)) {
4     echo "Tem Irix";
5 }
6 if (in_array("mac", $os)) {
7     echo "Tem mac";
8 }
9 ?>
```

<sup>6</sup>[https://www.php.net/manual/pt\\_BR/function.in-array.php](https://www.php.net/manual/pt_BR/function.in-array.php)

- A função **count(\$array)**<sup>7</sup> conta quantos elementos estão contidos em um array, e retorna essa quantidade

```
1 <?php
2 $a = array(1, 3, 5);
3 $result = count($a);
4 print ($result);
5
6 $b = array(7, 9, 11, 18);
7 $result = count($b);
8 print ($result);
9 ?>
```

---

<sup>7</sup>[https://www.php.net/manual/pt\\_BR/function.count.php](https://www.php.net/manual/pt_BR/function.count.php)



## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade

# Funções para String

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- Essas são algumas das funções para manipulação de Strings
  - string **rtrim(\$str)**
  - string **trim(\$str)**
  - string **strtolower(\$str)**
  - string **strtoupper(\$str)**
  - string **strrev(\$str)**
  - int **strlen(\$str)**
  - string **str\_replace(\$velho, \$novo, \$str)**
  - int **substr\_count(\$str, \$busca)**
  - string **htmlspecialchars(\$str)**

- A função **rtrim(\$str)**<sup>8</sup> remove os espaços em branco do final da string passada como parâmetro

```
1 <?php
2 $a = "Hello world!! \t";
3
4 var_dump($a);
5
6 echo "<br>";
7
8 var_dump(rtrim($a));
9 ?>
```

string(21) "Hello world!! "  
string(13) "Hello world!!"

<sup>8</sup>[https://www.php.net/manual/pt\\_BR/function.rtrim.php](https://www.php.net/manual/pt_BR/function.rtrim.php)

- A função **trim(\$str)**<sup>9</sup> remove os espaços em branco do início e do fim de uma string

```
1 <?php
2 $a = "    \t\tEsse é um texto com espaços    ";
3
4 var_dump($a);
5
6 echo "<br>";
7
8 var_dump(trim($a));
9 ?>
```

```
string(40) " Esse é um texto com espaços "
string(29) "Esse é um texto com espaços"
```

<sup>9</sup>[https://www.php.net/manual/pt\\_BR/function.trim.php](https://www.php.net/manual/pt_BR/function.trim.php)



- A função **strtolower(\$str)**<sup>10</sup> converte os caracteres da string passada como parâmetro para minúsculos

```
1 <?php
2 $str = "Mary Had A Little Lamb and She LOVED It So";
3 $str = strtolower($str);
4 print $str; // Prints mary had a little lamb and she loved it so
5 ?>
```

---

<sup>10</sup>[https://www.php.net/manual/pt\\_BR/function strtolower.php](https://www.php.net/manual/pt_BR/function strtolower.php)

- A função **strtoupper(\$str)**<sup>11</sup> converte os caracteres da string passada como parâmetro para maiúsculos

```
1 <?php
2 $str = "Mary Had A Little Lamb and She LOVED It So";
3 $str = strtoupper($str);
4 print $str; // Prints MARY HAD A LITTLE LAMB AND SHE LOVED IT SO
5 ?>
```

---

<sup>11</sup>[https://www.php.net/manual/pt\\_BR/function.strtoupper.php](https://www.php.net/manual/pt_BR/function.strtoupper.php)

- A função **strrev(\$str)**<sup>12</sup> retorna a string passada como parâmetro invertida

```
1 <?php
2 echo strrev("Hello world!"); // mostra "!dlrow olleH"
3 ?>
```

---

<sup>12</sup>[https://www.php.net/manual/pt\\_BR/function.strrev.php](https://www.php.net/manual/pt_BR/function.strrev.php)

- A função **strlen(\$str)**<sup>13</sup> retorna a quantidade de caracteres que compõem a string passada como parâmetro

```
1 <?php
2 $str = 'abcdef';
3 echo strlen($str); // 6
4 ?>
```

---

<sup>13</sup>[https://www.php.net/manual/pt\\_BR/function strlen.php](https://www.php.net/manual/pt_BR/function strlen.php)

- A função **str\_replace(\$velho, \$novo, \$str)**<sup>14</sup> busca pelo valor contido no parâmetro \$velho e substitui pelo valor contido em \$novo na string \$str

```

1  <?php
2  $frase = "você comeria frutas, vegetais, e fibra todos os dias.";
3
4  $frase2 = str_replace("comeria", "beberia", $frase);
5
6  print($frase2);
7
8  echo "<br>";
9
10 $saudavel = array("frutas", "vegetais", "fibra");
11 $saboroso  = array("pizza", "cerveja", "sorvete");
12
13 $novafrase = str_replace($saudavel, $saboroso, $frase);
14
15 print($novafrase);
16 ?>

```

<sup>14</sup>[https://www.php.net/manual/pt\\_BR/function.str-replace.php](https://www.php.net/manual/pt_BR/function.str-replace.php)

# substr\_count()

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- A função **substr\_count(\$str, \$busca [, \$offset [, \$length]])**<sup>15</sup> retorna o número de vezes que a substring foi encontrada. Note que a busca faz distinção de maiúsculas e minúsculas
- Além disso ainda podemos especificar somente uma parte da string para buscar utilizando os parâmetros \$offset e \$length

```
1 <?php
2 $text = 'This is a test';
3
4 echo substr_count($text, 'is'); // 2
5 ?>
```

<sup>15</sup>[https://www.php.net/manual/pt\\_BR/function.substr-count.php](https://www.php.net/manual/pt_BR/function.substr-count.php)

# htmlspecialchars()

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- A função **htmlspecialchars(\$str)**<sup>16</sup> retorna uma string transformando caracteres especiais em texto comum
- Ex: `<br>` => `&lt;br&rt;`
- Este tipo de alteração no texto evita um ataque chamado injeção de código

```
1 <?php
2 $new = htmlspecialchars("<a href='test'>Test</a>");
3 echo $new; // &lt;a href='test'&gt;Test&lt;/a&gt;
4 ?>
```

<sup>16</sup>[https://www.php.net/manual/pt\\_BR/function htmlspecialchars.php](https://www.php.net/manual/pt_BR/function htmlspecialchars.php)



## ① Escopo de variáveis

## ② Funções

## ③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

## ④ Validação de formulário

## ⑤ Atividade



- Algumas funções que são utilizadas para testar essas entradas em variáveis.
  - bool **isset(\$var)**
  - bool **is\_string(\$var)**
  - bool **is\_numeric(\$var)**
  - bool **is\_null(\$var)**

# isset()

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- A função **isset(\$var [, \$var])**<sup>17</sup> retorna *false* quando o valor de \$var for igual a **NULL**, caso contrário a função retornará *true*
- Se forem passadas mais de uma variável somente será retornado *true* caso todas as variáveis sejam diferentes de **NULL**

```
1 <?php
2 $var = '';
3 var_dump(isset($var)); // TRUE
4
5 $set = 1;
6 var_dump(isset($set)); // TRUE
7
8 var_dump(isset($bar)); // FALSE
9
10 $foo = NULL;
11 var_dump(isset($foo)); // FALSE
12 ?>
```

<sup>17</sup>[https://www.php.net/manual/pt\\_BR/function.isset.php](https://www.php.net/manual/pt_BR/function.isset.php)

- A função **is\_string(\$var)**<sup>18</sup> retorna verdadeiro caso a variável passada como parâmetro seja uma string

```
1 <?php
2 var_dump(is_string('abc')); // TRUE
3 var_dump(is_string("23")); // TRUE
4 var_dump(is_string(23.5)); // FALSE
5 var_dump(is_string(true)); // FALSE
6 ?>
```

<sup>18</sup>[https://www.php.net/manual/pt\\_BR/function.is-string.php](https://www.php.net/manual/pt_BR/function.is-string.php)

# is\_numeric()

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

- A função **is\_numeric(\$var)**<sup>19</sup> retornará *true* quando a variável passada como parâmetro for um número
- Todos os dados vindos do formulário sempre são textuais, portando mesmo quando uma string do tipo "123" é passada para a função a mesma retorna *true*

```
1 <?php
2 var_dump(is_numeric('123')); // TRUE
3 var_dump(is_numeric(23.7)); // TRUE
4 var_dump(is_numeric('abc')); // FALSE
5 var_dump(is_numeric(TRUE)); // FALSE
6 ?>
```

<sup>19</sup>[https://www.php.net/manual/pt\\_BR/function.is-numeric.php](https://www.php.net/manual/pt_BR/function.is-numeric.php)

- A função **is\_null(\$var)**<sup>20</sup> retorna *true* caso a variável passada como parâmetro seja **NULL**
- Essa função é o contrário da `isset()`

```
1 <?php
2 var_dump(is_null(NULL)); // TRUE
3 var_dump(is_null("")); // FALSE
4 var_dump(is_null(0)); // FALSE
5 var_dump(is_null("0")); // FALSE
6 ?>
```

<sup>20</sup>[https://www.php.net/manual/pt\\_BR/function.is-null.php](https://www.php.net/manual/pt_BR/function.is-null.php)



① Escopo de variáveis

② Funções

③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

④ Validação de formulário

⑤ Atividade

## Exemplo de formulário

**Contato**

Todos os campos são obrigatórios  
somente para maiores de 18 anos

**Nome:**

**E-mail:**

**Idade**

**Msg:**

**Enviar**

# Validação de formulário

Desenvolvimento  
de Aplicativo  
WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Escopo de  
variáveis

Funções

Funções do  
PHP

Debug

Arrays

String

Validação de  
variáveis

Validação de  
formulário

Atividade

```

1  <?php
2  if(isset($_GET['botao'])) {
3      $nome = $_GET['nome'];
4      $email = $_GET['email'];
5      $idade = $_GET['idade'];
6      $msg = $_GET['msg'];
7      $erro = array();
8      if (trim($nome) == NULL || trim($nome) == ""){
9          $erro[] = "O campo nome deve ser preenchido"; }
10     if (trim($email) == NULL || trim($email) == ""){
11         $erro[] = "O e-mail deve ser preenchido"; }
12     if (trim($idade) == NULL || trim($idade) == ""){
13         $erro[] = "A idade deve ser preenchida"; }
14     if (trim($msg) == NULL || trim($msg) == ""){
15         $erro[] = "A mensagem deve ser preenchida"; }
16     if ($idade < 18){
17         $erro[] = "Deve ter mais de 18 anos"; }
18     if (substr_count($email, "@") != 1){
19         $erro[] = "Email inválido"; }
20
21     if (count($erro) == 0) {
22         echo "Obrigado $nome,<br>Seus dados foram enviados com sucesso.";
23     } else {
24         echo "<pre>";
25         print_r($erro);
26         echo "</pre>";
27     }
28     ?>

```





① Escopo de variáveis

② Funções

③ Funções do PHP

Debug

Arrays

String

Validação de variáveis

④ Validação de formulário

⑤ Atividade

- Replique o formulário ao lado, em seguida faça a validação do mesmo considerando que todos os campos são obrigatórios, a pessoa deve possuir mais de 18 anos, os e-mails devem ser iguais, a pessoa é obrigada a aceitar os termos.
- Essa validação deve ser feita no PHP.



**Cadastre-se**

Todos os campos são obrigatórios  
somente para maiores de 18 anos

Nome:

Sexo Masc ☐ | Fem ☐

E-mail:

Confirme E-mail

Ano Nascimento

Cidade

UF

Aceito os termos ☐

## Atividade 2

### Desenvolvimento de Aplicativo WEB I

Prof. Me.  
Cleber  
Schroeder  
Fonseca

### Escopo de variáveis

### Funções

### Funções do PHP

Debug  
Arrays  
String  
Validação de  
variáveis

### Validação de formulário

### Atividade

- Verifique se os campos nome e e-mail foram informados.
- Verifique se o e-mail é válido (conforme o exemplo do slide 48).
- Se alguma das verificações acima não for validada deve ser mostrado qual foi o erro encontrado.

Nome

Email

Sexo  
☒ Masculino ☐ Feminino

Endereço

Número

Cidade

Estado

MUITO  
OBRIGADO!

Cleber Schroeder Fonseca

<http://ifrs.edu.br/riogrande>

[profcleberfonseca@gmail.com](mailto:profcleberfonseca@gmail.com)

[cleber.fonseca@riogrande.ifrs.edu.br](mailto:cleber.fonseca@riogrande.ifrs.edu.br)